Lightweight Hybrid Tableaux

Guillaume Hoffmann

Lightweight Hybrid Tableaux

Guillaume Hoffmann

INRIA Nancy Grand Est
Nancy, France

Abstract

We present a decision procedure for hybrid logic equipped with nominals, the satisfaction operator and existential, difference, converse, reflexive, symmetric and transitive modalities. This decision procedure is a prefixed tableau method based on the one introduced by Bolander and Blackburn in [2]. It enhances its predecessor in terms of computational efficiency and handles more expressive logics. Its way of ensuring termination enables addition of rules for the difference modality, inspired by Kaminski and Smolka in [5].

Keywords: Hybrid logic, tableau, loop-check, difference modality, converse modality

1 Hybrid Logic

In this article we consider two fragments of the hybrid language $\mathcal{H}(@, \mathsf{E}, \mathsf{D}, \Diamond^-)$ defined with signature $\mathsf{Sig} = \langle \mathsf{PROP}, \mathsf{NOM}, \mathsf{REL}, \mathcal{R}, \mathcal{T}, \mathcal{S} \rangle$ where PROP is a set of ordinary propositional symbols, NOM is a set of nominals and REL is a set of relational symbols, and $\mathcal{R}$, $\mathcal{S}$, $\mathcal{T}$ are subsets of REL. The sets PROP, NOM and REL are taken to be disjoint.

The hybrid language $\mathcal{H}(@, \mathsf{E}, \mathsf{D}, \Diamond^-)$ with signature Sig is defined by the following grammar:

$$\varphi \ ::= p \mid a \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Diamond_i\varphi \mid \Diamond_i^-\varphi \mid a\varphi \mid \mathsf{E}\varphi \mid \mathsf{D}\varphi$$

where $p \in \mathsf{PROP}$, $a \in \mathsf{NOM}$ and $i \in \mathsf{REL}$.

The main difference between hybrid logic and modal logic is the presence of nominals, which are propositional symbols true at *exactly one* world in a model; that is, a nominal "points to a unique world". Thus the formula $a\varphi$ is intended to express that the formula φ is true at the world pointed to by the nominal a. It is sometimes written $@_a\varphi$ and is called a satisfaction statement. We write $nom(\varphi)$ to denote the set of all nominals present in the formula φ. We have a fixed set of unary operators named $\Diamond_i$ for each $i \in \mathsf{REL}$ with their converses $\Diamond_i^-$. E is called the *existential modality* and D the *difference modality*.

We interpret these formulas on Kripke models:

Definition 1.1 (**Hybrid Model**) A *model* for $\mathcal{H}(@, \mathsf{E}, \mathsf{D}, \Diamond^-)$ with signature Sig is a tuple $(W, (R_i)_{i \in \mathsf{REL}}, V)$ where:

(i) W is a non-empty set of elements usually called *worlds* or *states*.

(ii) For all $i \in \mathsf{REL}$, R_i is a relation on W of arity 2; each R_i is called an *accessibility relation*. When $i \in \mathcal{R}$, the relation R_i is reflexive, when $i \in \mathcal{T}$ it is transitive and when $i \in \mathcal{S}$ it is symmetric.

(iii) For each proposition symbol or nominal s, $V(s)$ is a subset of W. If s is a nominal then $V(s)$ is a singleton set.

The relation $\mathcal{M}, w \models \varphi$ is defined inductively, where $\mathcal{M} = (W, (R_i)_{i \in \mathsf{REL}}, V)$ is a model, w is an element of W, and φ is a formula of our hybrid logic:

$$\begin{aligned}
\mathcal{M}, w &\models s \text{ iff } w \in V(s), \text{ where } s \text{ is a propositional symbol or a nominal} \\
\mathcal{M}, w &\models \neg\varphi \text{ iff not } \mathcal{M}, w \models \varphi \\
\mathcal{M}, w &\models \varphi \wedge \psi \text{ iff } \mathcal{M}, w \models \varphi \text{ and } \mathcal{M}, w \models \psi \\
\mathcal{M}, w &\models \Diamond_i \varphi \text{ iff for some } v \in W,\ (w, v) \in R_i \text{ and } \mathcal{M}, v \models \varphi \\
\mathcal{M}, w &\models \Diamond_i^- \varphi \text{ iff for some } v \in W, (v, w) \in R_i \text{ and } \mathcal{M}, v \models \varphi \\
\mathcal{M}, w &\models a\varphi \text{ iff } \mathcal{M}, v \models \varphi, \text{ where } V(a) = \{v\} \\
\mathcal{M}, w &\models \mathsf{E}\varphi \text{ iff for some } v \in W,\ \mathcal{M}, v \models \varphi \\
\mathcal{M}, w &\models \mathsf{D}\varphi \text{ iff for some } v \in W,\ v \neq w \text{ and } \mathcal{M}, v \models \varphi
\end{aligned}$$

In what follows, the metavariables φ, ψ, χ, ... range over formulas, $p, q, r, \ldots$ range over ordinary propositional symbols, and $a, b, c, \ldots$ range over nominals.

The modality E can define satisfaction statements, since $@_a\varphi$ can be replaced by $\mathsf{E}(a \wedge \varphi)$. Also D can define $\mathsf{E}\varphi$ by $\varphi \vee \mathsf{D}\varphi$. As a consequence, we will no longer write $\mathcal{H}(@, \mathsf{E}, \mathsf{D})$ and $\mathcal{H}(@, \mathsf{E}, \mathsf{D}, \Diamond^-)$, but $\mathcal{H}(\mathsf{D})$ and $\mathcal{H}(\mathsf{D}, \Diamond^-)$. Moreover, we call Sig the full signature $\langle \mathsf{PROP}, \mathsf{NOM}, \mathsf{REL}, \mathcal{R}, \mathcal{T}, \mathcal{S} \rangle$ and Sig' the signature $\langle \mathsf{PROP}, \mathsf{NOM}, \mathsf{REL}, \mathcal{R}, \mathcal{T} \rangle$, i.e., a signature that does not specify symmetric relations.

2 A prefixed tableau calculus

We now present a prefixed tableau calculus for the hybrid language $\mathcal{H}(\mathsf{D}, \Diamond^-)$. Formulas occurring in the tableau rules are *prefixed formulas* of the form $\sigma\varphi$, where φ is a formula of $\mathcal{H}(\mathsf{D}, \Diamond^-)$ and σ belongs to some fixed countably infinite set of symbols PREF called *prefixes*. This set is disjoint from NOM and PROP and is well-ordered. Later, we will use the term "smallest prefix" and write $\sigma < \tau$, where σ and τ are prefixes, to refer to this well-order. The intended interpretation of a prefixed formula $\sigma\varphi$ is that σ denotes a world at which φ holds.

In addition to prefixed formulas, the tableau rules contain *accessibility statements* of the form $\sigma\Diamond_i\tau$ where σ and τ are prefixes. The intended interpretation of $\sigma\Diamond_i\tau$ is that the world denoted by τ is accessible from the world denoted by σ by the accessibility relation R_i.

A *tableau* in this calculus is a wellfounded, finitely branching tree in which each node is labelled by a prefixed formula, and the edges represent applications of tableau rules in the usual way. In the following we use the term *formula* to denote either a formula of $\mathcal{H}(\mathsf{D}, \Diamond^-)$ or a prefixed formula. When a prefixed formula $\sigma\varphi$ occurs in a tableau branch Θ, we write $\sigma\varphi \in \Theta$, and say that φ is true at σ on Θ, or that σ makes φ true on Θ. We call $\mathsf{Tab}(\varphi)$ the tableau that initially contains nodes labelled by the following prefixed formulas:

- $\sigma_0\varphi$, with σ_0 being a fresh prefix
- $\sigma(n)n$, with $\sigma(n)$ being a fresh prefix for each nominal $n \in nom(\varphi)$.

$\sigma_0\varphi$ is called the root formula, and $\sigma_0, \sigma(n)$ (for all $n \in nom(\varphi)$) are called *root prefixes*.

A *saturated tableau* is a tableau in which no more rules can be applied that satisfy the saturation constraints. A *saturated branch* is a branch of a saturated tableau. A branch of a tableau is called *closed* if it contains formulas $\sigma\varphi$ and $\sigma\neg\varphi$ for some σ and φ. Otherwise the branch is called *open*. A *closed tableau* is one in which all branches are closed, and an *open tableau* is one in which at least one branch is open.

Figure 1 presents the rules needed to handle the hybrid language $\mathcal{H}(\mathsf{D}, \Diamond^-)$ but the rules of Figure 2 are also needed to handle reflexive, transitive and symmetric modalities. In the rest of the article, we consider two subsystems of the calculus presented on Figure 1 and Figure 2 where only a subset of the rules are allowed to be applied. In such subsystems, a *saturated tableau* is a tableau in which none of the rules in the subset can be applied while satisfying the saturation conditions.

2.1 Tableau rules

Alongside with the definitions of rules given in Figure 1 and Figure 2, we introduce a few notions to complete the description of the calculus.

The definition of rule (νId) involves *local formulas*, let us define them:

Definition 2.1 A *local formula* is a formula of the shape s, $\neg s$, $\Diamond_i\varphi$, $\neg\Diamond_i\varphi$, $\Diamond_i^-\varphi$, $\neg\Diamond_i^-\varphi$, with s a propositional symbol or nominal.

The presence of nominals enables hybrid logic to express equality of worlds of the model. In order to handle this notion of equality, we manipulate equivalence classes of prefixes and representatives of these classes:

Definition 2.2 We define the binary relation $\sim_\Theta$ on the prefixes in a branch Θ by $\{(\sigma, \tau) \mid \sigma a, \tau a \in \Theta, a \in \mathsf{NOM}\}$. Note that this relation is reflexive.

Definition 2.3 Let Θ be a branch of a tableau, and let σ be a prefix occurring in Θ. The *nominal urfather* of σ on Θ, written $n_\Theta(\sigma)$, is defined to be the smallest prefix τ in Θ for which $\tau \sim_\Theta \sigma$. A prefix σ is called a *nominal urfather* in Θ if $\sigma = n_\Theta(\tau)$ for some prefix τ.

Later, it will come clear that the relation $\sim_\Theta$ defines an equivalence class of prefixes, and we will show that the nominal urfather is uniquely defined for a given class, and serves as its representative.

$$\frac{\sigma\neg\neg\varphi}{\sigma\varphi}\,(\neg\neg) \qquad \frac{\sigma\varphi}{n_\Theta(\sigma)\varphi}\,(\nu Id)^1 \qquad \frac{\sigma\Diamond\tau}{u_\Theta(\sigma)\Diamond u_\Theta(\tau)}\,(bridge)$$

$$\frac{\sigma(\varphi\wedge\psi)}{\sigma\varphi,\sigma\psi}\,(\wedge) \qquad \frac{\sigma\Diamond_i\varphi}{\sigma\Diamond_i\tau,\tau\varphi}\,(\Diamond)^2 \qquad \frac{\sigma a\varphi}{\sigma(a)\varphi}\,(@)$$

$$\frac{\sigma\neg(\varphi\wedge\psi)}{\sigma\neg\varphi \mid \sigma\neg\psi}\,(\neg\wedge) \qquad \frac{\sigma\neg\Diamond_i\varphi,\sigma\Diamond_i\tau}{\tau\neg\varphi}\,(\neg\Diamond) \qquad \frac{\sigma\neg a\varphi}{\sigma(a)\neg\varphi}\,(\neg@)$$

$$\frac{\sigma\mathsf{E}\varphi}{\tau\varphi}\,(\mathsf{E})^2 \qquad \frac{\sigma\Diamond_i^-\varphi}{\tau\Diamond_i\sigma,\tau\varphi}\,(\Diamond^-)^2 \qquad \frac{\sigma\mathsf{D}\varphi}{\sigma\neg n,\tau n,\tau\varphi}\,(\mathsf{D})^{2,4}$$

$$\frac{\sigma\neg\mathsf{E}\varphi}{\gamma\neg\varphi}\,(\neg\mathsf{E})^3 \qquad \frac{\sigma\neg\Diamond_i^-\varphi,\tau\Diamond_i\sigma}{\tau\neg\varphi}\,(\neg\Diamond^-) \qquad \frac{\sigma\neg\mathsf{D}\varphi}{\sigma n,\gamma n \mid \gamma\varphi}\,(\neg\mathsf{D})^{3,4}$$

[1] φ is a local formula or an accessibility statement.
[2] The prefix τ is new to the branch.
[3] The prefix γ is already in the branch.
[4] The nominal n is new to the branch.

Saturation constraints:

- a formula is never added to a tableau branch where it already occurs. Thus, (@) and (¬@) are never applied to $\sigma\varphi$ if they have already been applied to $\tau\varphi$.
- $(\Diamond^{(-)})$ can not be applied to $\sigma\varphi$ on Θ if it has been applied to $\tau\varphi$ with $\sigma\sim_\Theta\tau$
- (E) is never applied to $\sigma\mathsf{E}\varphi$ if there is a prefix τ such that $\tau\varphi$.
- (D) is never applied to $\sigma\mathsf{D}\varphi$ if there is a prefix τ such that $\tau\varphi$ without $\sigma\sim_\Theta\tau$.
- (¬D) is never applied to $\sigma\neg\mathsf{D}\varphi$ and γ if it has already been applied to $\tau\neg\mathsf{D}\varphi$ and γ with $\sigma\sim_\Theta\tau$.

Fig. 1. Prefixed tableau calculus for the hybrid language $\mathcal{H}(\mathsf{D},\Diamond^-)$.

We mentioned earlier that the set PREF was well-ordered. Rules that introduce new prefixes on the branch – i.e. $(\Diamond)$, $(\Diamond^-)$, (E), (D) – pick the smallest prefix of PREF that is not already in the branch. We sometimes call applications of the rules $(\Diamond)$ and $(\Diamond^-)$ *diamond expansions.*

2.2 Quasi-subformula property

As tableaux systems are about taking apart formulas, they usually manipulate only subformulas of a given input formula. In the present case we want to show a similar property, which will later enable us to prove termination.

$$\frac{\sigma\neg\Diamond_i\varphi}{\sigma\neg\varphi}\,(re), i \in \mathcal{R}$$

$$\frac{\sigma_0\Diamond_i\sigma_1, \sigma_0\neg\Diamond_i\varphi}{\sigma_1\neg\Diamond_i\varphi}\,(tr), i \in \mathcal{T} \qquad \frac{\sigma_0\Diamond_i\sigma_1, \sigma_1\neg\Diamond_i^-\varphi}{\sigma_0\neg\Diamond_i^-\varphi}\,(tr^-), i \in \mathcal{T}$$

$$\frac{\sigma\neg\Diamond_i\varphi}{\sigma\neg\Diamond_i^-\varphi}\,(sy), i \in \mathcal{S} \qquad \frac{\sigma\neg\Diamond_i^-\varphi}{\sigma\neg\Diamond_i\varphi}\,(sy^-), i \in \mathcal{S}$$

Fig. 2. Rules for reflexive, transitive and symmetric modalities.

Definition 2.4 Given a tableau branch Θ and a prefix σ, the *set of true formulas* at σ on Θ, written $T^\Theta(\sigma)$ is

$$T^\Theta(\sigma) = \{\varphi \mid \sigma\varphi \in \Theta\}.$$

Definition 2.5 A formula φ is said to be a *quasi-subformula* of a formula ψ if one of the following holds:

- φ is a subformula of ψ.
- φ has the form $\neg\chi$, where χ is a subformula of ψ.

Lemma 2.6 (Quasi-subformula Property) *Let Θ be a branch of a tableau built with the prefixed formula $\sigma_0\varphi_0$ as root. For any prefixed formula $\sigma\varphi$ occurring on Θ such that φ is not a (negated) nominal generated by (D) or $(\neg D)$, φ is a quasi-subformula of φ_0.*

Proof. This is easily seen by going through each rule of Figure 1 and Figure 2. □

Corollary 2.7 *Let Θ be a branch and σ a prefix in Θ. Let $T^\Theta_{sub}(\sigma)$ be $T^\Theta(\sigma)$ without the (negated) nominals generated by (D) and $(\neg D)$. The set $T^\Theta_{sub}(\sigma)$ is finite.*

We will now establish soundness and completeness of two subsystems with regards to their semantics, and prove their termination. We let the reader verify that soundness is preserved, by the usual argument of satisfiability-preserving branch expansion.

3 Decision procedure for $\mathcal{H}(\mathbf{D})$ with reflexive and transitive modalities

In this section, we consider the language $\mathcal{H}(\mathsf{D})$ over the signature Sig' given by the following grammar:

$$\varphi \ ::= \ p \mid a \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Diamond_i\varphi \mid a\varphi \mid \mathsf{E}\varphi \mid \mathsf{D}\varphi$$

where $p \in \mathsf{PROP}$, $a \in \mathsf{NOM}$, $i \in \mathsf{REL}$ and $\mathcal{R} \subseteq \mathsf{REL}$ and $\mathcal{T} \subseteq \mathsf{REL}$.

The expression *the calculus of* $\mathcal{H}(D)$ refers to the calculus consisting of the rules:

$$(\neg\neg), (\wedge), (\neg\wedge), (\Diamond), (\neg\Diamond), (re), (tr), (\nu Id), (@), (\neg@), (\mathsf{E}), (\neg\mathsf{E}), (\mathsf{D}), (\neg\mathsf{D})$$

3.1 Termination

The general idea of the termination proof of this system is that on one hand, the quasi-subformula property guarantees that a given prefix can only make true finitely many formulas, and on the other hand, the restriction (or loop-check) that we introduce guarantees that there can only be finitely many prefixes in a branch. As a consequence, the number of formulas, and hence of rule applications, can only be finite, which means that the calculus is guaranteed to terminate.

Definition 3.1 For a prefix σ, let $L^{\Theta}(\sigma)$ be the set of formulas true at $n_{\Theta}(\sigma)$, of the shape $\Diamond\varphi$, $\Diamond^{-}\varphi$, $\neg\Diamond\varphi$, $\neg\Diamond^{-}\varphi$, s and $\neg s$, with s being a propositional symbol or a nominal not introduced by the rule $(\neg\mathsf{D})$. We call these formulas *model-relevant local formulas.*

Now for the first important result:

Lemma 3.2 *Let σ be a prefix occurring in a branch Θ in the calculus of $\mathcal{H}(D)$. The set $L^{\Theta}(\sigma)$ is finite.*

Proof. Only a finite number of formulas of the shape $\mathsf{D}\varphi$ can appear in a branch, and (D) can be applied at most twice for each one of them as premise. This and Corollary 2.7 prove this theorem. □

We are going to divide prefixes in two categories: those who "can generate new prefixes" and those who cannot, or the blocked ones. During the run of the calculus, prefixes of a branch can pass from one category to the other. Ultimately, we will see that there can be only a finite number of prefixes belonging to the first category, which will imply termination of the calculus. A prefix σ will belong to the blocked category when its L^{Θ} set is included in the L^{Θ} set of a smaller prefix. As a consequence, diamond expansions of formulas $\sigma\Diamond\varphi$ are forbidden.

Definition 3.3 We define the *inclusion urfather* of a prefix σ in a branch Θ, written $i_{\Theta}(\sigma)$, to be the smallest prefix τ for which: $L^{\Theta}(\sigma) \subseteq L^{\Theta}(\tau)$. A prefix σ is called an *inclusion urfather* in Θ if $\sigma = i_{\Theta}(\tau)$ for some prefix τ.

Definition 3.4 (**Loop-check** $(\mathcal{I})$) The rule $(\Diamond)$ is only applied to a formula $\sigma\varphi$ on a branch if σ is an inclusion urfather on that branch.

One consequence of this loop-check is that a formula $\Diamond\varphi$ has to be copied to the nominal urfather of its prefix before being expanded.

Lemma 3.5 *Let Θ be a branch in the calculus of $\mathcal{H}(D)$ with finitely many prefixes in it, and σ a prefix occurring in it. $T^{\Theta}(\sigma)$ is finite.*

Proof. Note that according to our tableau conventions, all prefixed formulas occurring in the infinite branch Θ are distinct.

As seen in Lemma 2.6, we know that there are finitely many quasi-subformulas of the input formula in the branch. So, by the quasi-subformula property, we know

that a calculus that does not involve (D) nor (¬D) generates finitely many formulas for a given prefix.

Now let us see whether there are also finitely many formulas generated by the rules (D) and (¬D). Since there are finitely many prefixes in Θ, and by the quasi-subformula property, there are finitely many formulas of the shape $\sigma\mathsf{D}\varphi$ or $\sigma\neg\mathsf{D}\varphi$ in the branch. Moreover, as (D) can at most be fired twice for a given premise $\mathsf{D}\varphi$, there are finitely many formulas generated by (D). Similarly, by the saturation condition of (¬D), and as the number of prefixes in the branch is finite, there are finitely many applications of (¬D) rules for a given formula $\sigma\neg\mathsf{D}\varphi$. This also means that there are finitely many formulas generated by (¬D). Since all the formulas generated by (D) and (¬D) are either (negated) nominals or quasi-subformulas of the input formula, there is also a finite number of rule applications caused by conclusions of these rules. □

Theorem 3.6 *Any tableau in the calculus of $\mathcal{H}(D)$ constructed under restriction $(\mathcal{I})$ is finite.*

Proof. The prefixes present in a branch are either root prefixes, prefixes introduced by (E) and (D), or prefixes introduced by (◊). We already know that there are finitely many root prefixes. Moreover, as only a finite number of subformulas of the input formula are of the shape $\mathsf{E}\varphi$ or $\mathsf{D}\varphi$, the saturation condition of (E) and (D) ensure that only a finite number of prefixes can be generated by these rules. Let us now consider the number of diamond expansions:

- because of Lemma 3.2 and the loop-check $(\mathcal{I})$, the maximal number of equivalence classes in which prefixes are allowed to expand diamonds is 2^N, where $N = |\{\varphi \mid \varphi$ model-relevant local formula $\}|$
- because of saturation of (◊), the maximal number of diamond expansions in a given equivalence class is $M = |\{\Diamond\varphi \mid \varphi$ quasi-subformula of the input formula$\}|$

Thus the number of prefixes generated by (◊) in a branch is bounded by $M \times 2^N$.

As there can be only finitely many prefixes in a branch, the result follows by Lemma 3.5. □

3.2 Completeness

We will now prove that the calculus of $\mathcal{H}(\mathsf{D})$ is complete. For this, we need a certain amount of properties about nominal urfathers and inclusion urfathers, so that a saturated open branch has the desirable properties. We first need to take care of nominal urfathers, which are privileged prefixes because of the (νId) rule, and then we focus on inclusion urfathers, given that they are used to build a model from a saturated open branch. Let us start by some properties of nominal urfathers:

Lemma 3.7 (Nominal Urfather Equality) *Let Θ be a saturated branch in a calculus containing at least (νId). If $\sigma \sim_\Theta \tau$ then $n_\Theta(\sigma) = n_\Theta(\tau)$.*

Proof. Assume $\sigma \sim_\Theta \tau$. Then there is a nominal a such that $\sigma a, \tau a \in \Theta$. By saturation by the (νId) rule, $n_\Theta(\sigma)a$ and $n_\Theta(\tau)a$ hold. Suppose $n_\Theta(\sigma) \neq n_\Theta(\tau)$. Without loss of generality, suppose $n_\Theta(\sigma) < n_\Theta(\tau)$. Then, a would be a nominal

true at τ and $n_\Theta(\sigma)$, which contradicts the assumption that $n_\Theta(\tau)$ is the smallest prefix such that there is a nominal true at τ and $n_\Theta(\tau)$. □

Note that the root prefix σ_0 of a tableau branch is always a nominal urfather on that branch. More generally, any prefix σ for which $n_\Theta(\sigma) = \sigma$ is a nominal urfather on Θ. The other direction also holds, as the following lemma shows:

Lemma 3.8 (Nominal Urfather Characterisation) *Let Θ be a saturated branch in a calculus containing at least (νId). Then σ is a nominal urfather on Θ if and only if $n_\Theta(\sigma) = \sigma$.*

Proof. Let us consider the 'only if' direction. If σ is a nominal urfather then $n_\Theta(\tau) = \sigma$ for some prefix τ. If $\tau = \sigma$ then the proof is complete. Otherwise $\sigma \sim_\Theta \tau$, by definition of n_Θ. Then, by Urfather Equality (Lemma 3.7), $n_\Theta(\sigma) = n_\Theta(\tau) = \sigma$.□

We need now to prove some results concerning inclusion urfathers. The first result enables us to claim in some cases that nominal and inclusion urfather are the same:

Lemma 3.9 *Let Θ be a saturated branch in a calculus containing at least (νId). If σ is a prefix making at least one model-relevant nominal true on Θ then the nominal urfather and the inclusion urfather of σ coincide.*

Proof. Assume $\sigma a \in \Theta$, with a being a model-relevant nominal. We need to prove $n_\Theta(\sigma) = i_\Theta(\sigma)$. Closure under the (νId) rule gives us that $n_\Theta(\sigma)a \in \Theta$, so $a \in L^\Theta(\sigma)$. Let $\tau = i_\Theta(\sigma)$. By definition, τ is the smallest prefix such that $L_\Theta(\sigma) \subseteq L_\Theta(\tau)$, so $a \in L_\Theta(\tau)$. Hence $n_\Theta(\tau)a \in \Theta$.

Assume $\tau \neq n_\Theta(\sigma)$. The case $\tau > n_\Theta(\sigma)$ is impossible, because then $n_\Theta(\sigma)$ would be a candidate inclusion urfather of σ smaller than τ, by closure under (νId). So let us assume $\tau < n_\Theta(\sigma)$. As $n_\Theta(\tau) \leq \tau$, we then have $n_\Theta(\tau) < n_\Theta(\sigma)$, but since $n_\Theta(\tau)$ makes the nominal a true on Θ, this contradicts the fact that $n_\Theta(\sigma)$ is the nominal urfather of σ. □

We have proved two basic properties for nominal urfathers: Nominal Urfather Equality (Lemma 3.7) and Nominal Urfather Characterisation (Lemma 3.8). We are going to see that these properties also hold for inclusion urfathers.

Lemma 3.10 (Inclusion Urfather Equality) *Let Θ be a saturated branch in a calculus containing at least (νId). If $\sigma \sim_\Theta \tau$, then $i_\Theta(\sigma) = i_\Theta(\tau)$.*

Proof. σ and τ have the same nominal urfather (Lemma 3.7), thus they have the same inclusion urfather. □

Lemma 3.11 (Inclusion Urfather Characterisation) *Let Θ be a saturated branch in a calculus containing at least (νId). Then σ is an inclusion urfather on Θ if and only if $i_\Theta(\sigma) = \sigma$.*

Proof. For the "only if" direction, suppose σ is an inclusion urfather, i.e., there exists a prefix τ such that $\sigma = i_\Theta(\tau)$. Let us show that $i_\Theta(\sigma) = \sigma$. By definition, σ is the smallest prefix such that $L^\Theta(\tau) \subseteq L^\Theta(\sigma)$ for a prefix τ. Suppose that there is a prefix $\gamma = i_\Theta(\sigma)$ and $\gamma < \sigma$. Therefore, $L^\Theta(\tau) \subseteq L^\Theta(\sigma) \subseteq L^\Theta(\gamma)$, which contradicts the fact that σ is the inclusion urfather of τ. □

And a third property is going to be useful:

Lemma 3.12 *Given a saturated branch Θ in a calculus containing at least (νId) and a prefix σ, $n_\Theta(i_\Theta(\sigma)) = i_\Theta(\sigma)$.*

Proof. Let $\tau = i_\Theta(\sigma)$. Assume $n_\Theta(\tau) \neq \tau$. Necessarily, $n_\Theta(\tau) < \tau$. Since by nominal Urfather Characterisation (Lemma 3.8), $n_\Theta(n_\Theta(\tau)) = n_\Theta(\tau)$, and by saturation by (νId), $n_\Theta(\tau)$ is also a candidate to be the inclusion urfather of σ, and since it is smaller than τ, we have a contradiction. □

Lemma 3.13 (Inclusion Urfather Closure) *Let Θ be a saturated branch in a calculus containing at least (νId). If Θ contains $\sigma\varphi$ with φ a model-relevant local formula, then Θ contains $i_\Theta(\sigma)\varphi$.*

Proof. By saturation of (νId) and definition of an inclusion urfather, $n_\Theta(i_\Theta(\sigma))\varphi \in \Theta$ which gives us $i_\Theta\varphi \in \Theta$ by Lemma 3.12. □

We are now ready to build a model from a saturated open branch and prove a correspondence between formulas in the branch and truth in the model. Given an open, saturated branch Θ with root $\sigma_0\varphi_0$, we define a model $\mathcal{M}^\Theta$ by

$$\begin{aligned}
\mathcal{M}^\Theta &= (W^\Theta, (R_i^\Theta)_{i<n}, V^\Theta), \text{ where} \\
W^\Theta &= \{\sigma \mid \sigma \text{ is an inclusion urfather on } \Theta\} \\
R_i^\Theta &= \{(\sigma, i_\Theta(\tau)) \mid \sigma \in W^\Theta \text{ and } \sigma\Diamond_i\tau \text{ occurs on } \Theta\} \\
V^\Theta(s) &= \{i_\Theta(\sigma) \mid \sigma s \text{ occurs on } \Theta\}.
\end{aligned}$$

In this definition, s is a propositional symbol or a nominal. Inclusion Urfather Equality (Lemma 3.10) implies that $V^\Theta(a)$ is a singleton for any nominal a.

We then define the model $\mathcal{M}_*^\Theta$ as $\mathcal{M}^\Theta$ in which the missing links for reflexive and transitive relations have been added, that is. For every relation R_i in $\mathcal{M}^\Theta$, we write R_{i*} for its reflexive closure when $i \in \mathcal{R}$, its transitive closure when $i \in \mathcal{T}$ and its reflexive-transitive closure when $i \in \mathcal{R} \cap \mathcal{T}$. Otherwise R_{i*} is equal to R_i.

Lemma 3.14 *Let Θ be a saturated open branch built from the root formula ϕ in the calculus of $\mathcal{H}(D)$ with restriction $(\mathcal{I})$. For any formula $\sigma\varphi \in \Theta$ such that $nom(\varphi) \subseteq nom(\phi)$, we have $\mathcal{M}_*^\Theta, i_\Theta(\sigma) \models \varphi$.*

Proof. The proof is by induction on the syntactic structure of φ.

- $\varphi = p$. By definition, $i_\Theta(\sigma) \in V^\Theta(p)$. This implies $\mathcal{M}_*^\Theta, i_\Theta(\sigma) \models p$.
- $\varphi = \neg p$. By Urfather Closure, $i_\Theta(\sigma)\neg p \in \Theta$. Since Θ is open, $i_\Theta(\sigma)p \notin \Theta$. Thus $i_\Theta(\sigma) \notin V^\Theta(p)$, which implies $\mathcal{M}_*^\Theta, i_\Theta(\sigma) \models \neg p$.
- $\varphi = a$. By definition $V^\Theta(a) = \{i_\Theta(\sigma)\}$, therefore $\mathcal{M}_*^\Theta, i_\Theta(\sigma) \models a$.
- $\varphi = \neg a$. By Urfather Closure, we get $i_\Theta(\sigma)\neg a \in \Theta$. Since Θ contains initially a formula $\sigma_a a$, we have $V^\Theta(a) = \{i_\Theta(\sigma_a)\}$ as a is an input nominal. By Urfather Closure, we have $i_\Theta(\sigma_a)a \in \Theta$. Since Θ is an open branch containing both $i_\Theta(\sigma)\neg a$ and $i_\Theta(\sigma_a)a$, we get $i_\Theta(\sigma) \neq i_\Theta(\sigma_a)$. Thus we have $i_\Theta(\sigma) \notin V^\Theta(a)$ which implies $\mathcal{M}_*^\Theta, i_\Theta(\sigma) \models \neg a$.

- $\varphi = \neg\neg\psi$, $\varphi = \psi \wedge \chi$ and $\varphi = \neg(\psi \wedge \chi)$ are trivial, by application of the corresponding tableau rules and the induction hypothesis.
- $\varphi = a\psi$. By assumption, a can not be a nominal introduced by the calculus. By closure under the rules (@) and (νId), Θ must also contain $\tau\psi$, with τ being the smallest prefix such that τa occurs. Induction hypothesis gives us $\mathcal{M}^\Theta, i_\Theta(\tau) \models \psi$. Since τ is the smallest prefix making a nominal true, it is a nominal urfather, and as it makes one input nominal true, it is also an inclusion urfather (Lemma 3.9), so it is in the model. By Urfather Characterisation, $i_\Theta(\tau) = \tau$, and since $\tau a \in \Theta$, we get $V^\Theta(a) = \{\tau\}$. Thus, as we have $\mathcal{M}^\Theta, \tau \models \psi$ and $\mathcal{M}^\Theta, \tau \models a$, we have that for all $w \in W^\Theta$, $\mathcal{M}^\Theta, w \models a\psi$, in particular for $w = i_\Theta(\sigma)$, as needed.
- $\varphi = \neg a\psi$ closure by ($\neg$@) then similar to the previous case.
- $\varphi = \mathsf{E}\psi$. Closure under the (E) rule implies that Θ must also contain a formula $\tau\psi$ for some prefix τ. The induction hypothesis then gives us $\mathcal{M}^\Theta, i_\Theta(\tau) \models \psi$ which proves that $\mathcal{M}^\Theta, w \models \mathsf{E}\psi$ for all $w \in W^\Theta$, including $w = i_\Theta(\sigma)$.
- $\varphi = \neg\mathsf{E}\psi$. We need to prove $\mathcal{M}^\Theta, i_\Theta(\sigma) \models \neg\mathsf{E}\psi$, that is, for all $\tau \in W^\Theta$, $\mathcal{M}^\Theta, \tau \models \neg\psi$. Choose an arbitrary element τ in W^Θ. By closure under the ($\neg\mathsf{E}$) rule we have that $\tau\neg\psi$ occurs on Θ. The induction hypothesis gives us $\mathcal{M}^\Theta, i_\Theta(\tau) \models \neg\psi$. By Urfather Characterisation, we have $i_\Theta(\tau) = \tau$, thus $\mathcal{M}^\Theta, \tau \models \neg\psi$ as required.
- $\varphi = \mathsf{D}\psi$. Closure under the (D) rule implies that Θ also contains $\sigma\neg n$, τn and $\tau\psi$. As $\sigma\neg n$ and τn occur, by Urfather Closure we have $i_\Theta(\sigma)\neg n$ and $i_\Theta(\tau)n$, so, as the branch is open, $i_\Theta(\sigma) \neq i_\Theta(\tau)$. Moreover, as $\tau\psi \in \Theta$, then by induction hypothesis, $\mathcal{M}^\Theta, i_\Theta(\tau) \models \psi$. With $i_\Theta(\sigma) \neq i_\Theta(\tau)$, this means we have $\mathcal{M}^\Theta, i_\Theta(\sigma) \models \mathsf{D}\psi$.
- $\varphi = \neg\mathsf{D}\psi$. If there is no world $\tau \neq i_\Theta(\sigma)$ then this trivially holds. Otherwise, let τ be such a world. By saturation of ($\neg\mathsf{D}$), we have that either the formulas σn and τn are in Θ, or $\tau\neg\psi$ is. In the first case, $\sigma \sim_\Theta \tau$, which implies by Urfather Equality that $i_\Theta(\sigma) = i_\Theta(\tau)$. Thus, by Urfather Characterisation, $i_\Theta(\sigma) = \tau$, which is a contradiction. Now assume $\tau\neg\psi \in \Theta$. Then, by induction hypothesis and Urfather Characterisation, $\mathcal{M}^\Theta, \tau \models \neg\psi$, which is what we needed.
- $\varphi = \Diamond_i\psi$. By Inclusion Urfather Closure and saturation of ($\Diamond$), we have:

  $$i_\Theta(\sigma)\Diamond_i\tau,\ \tau\psi\ \in \Theta$$

  Then, by definition of R_i^Θ and induction hypothesis:

  $$(i_\Theta(\sigma), i_\Theta(\tau)) \in R_i^\Theta \quad , \quad \mathcal{M}_*^\Theta, i_\Theta(\tau) \models \psi$$

  Combining this, we obtain as required $\mathcal{M}_*^\Theta, i_\Theta(\sigma) \models \Diamond_i\psi$.
- $\varphi = \neg\Diamond_i\psi$. If there is no world σ_1 such that $(i_\Theta(\sigma), \sigma_1) \in R_{i*}^\Theta$ then this holds trivially. Otherwise, let such σ_1 be chosen such that $(i_\Theta(\sigma), \sigma_1) \in R_{i*}^\Theta$. We need to consider two subcases:
  · $(i_\Theta(\sigma), \sigma_1) \in R_i^\Theta$. By definition of R_i^Θ there must be a prefix τ_1 such that $\sigma_1 = i_\Theta(\tau_1)$ and $i_\Theta(\sigma)\Diamond_i\tau_1 \in \Theta$. Then by Urfather Closure, $i_\Theta(\sigma)\neg\Diamond_i\psi \in \Theta$, and by closure under ($\neg\Diamond$), $\tau_1\neg\psi \in \Theta$. Induction hypothesis entails $\mathcal{M}^\Theta, i_\Theta(\tau_1) \models \neg\psi$, i.e., $\mathcal{M}^\Theta, \sigma_1 \models \neg\psi$. From this it follows that $\mathcal{M}^\Theta, i_\Theta(\sigma) \models \neg\Diamond_i\psi$.

· $(i_\Theta(\sigma), \sigma_1) \in R^\Theta_{i*} \setminus R^\Theta_i$. If $i \in \mathcal{R}$ and $\sigma_1 = i_\Theta(\sigma)$, saturation by the rule (re) enforces the presence of $\neg\psi$ at the prefix $i_\Theta(\sigma)$, thus it follows that $\mathcal{M}^\Theta, i_\Theta(\sigma) \models \neg\Diamond_i\psi$. If $i \in \mathcal{T}$, saturation by the rule (tr) gives us $\sigma_1\neg\psi$.

□

Theorem 3.15 *The calculus of $\mathcal{H}(D)$ with restriction $(\mathcal{I})$ is complete.*

Proof. The prefix σ_0 of the root formula is an inclusion urfather. □

3.3 Discussion

We would like now to discuss some of the similarities and differences between the calculus we presented in this article and related work. In particular we will discuss the work of Bolander and Blackburn, and Kaminski and Smoka on hybrid tableaux calculi. Bolander and Blackburn introduced in [2], the first terminating tableau system for the basic hybrid logic $\mathcal{H}(@)$. For this language, both a prefixed and an internalised calculus were introduced. Moreover, they introduced a prefixed calculus for $\mathcal{H}(@, \mathsf{E}, \Diamond^-)$. Kaminski and Smolka introduced an internalised calculus for $\mathcal{H}(\mathsf{D})$ with reflexive and transitive relations in [4], and later extended it so as to handle the hybrid logic $\mathcal{H}(\mathsf{D}, \Diamond^-)$ [5].

Kaminski and Smolka's calculus and the difference modality: Kaminski and Smolka presented the first decision procedure for hybrid logic with the difference modality in [4]. Their calculus is internalised, for it is expressed in simple type theory, and equality and disequality are represented natively in their formalism with the symbols $=$ and $\neq$. In contrast, our calculus is prefixed, and we use new nominals to enforce equality and disequality respectively needed by the rules $(\neg\mathsf{D})$ and (D), which we adapted from their work.

Loop-check: The loop-check we use is also known as "subset blocking" and "anywhere blocking", notably in description logics tableau systems [1]. It is the same loop-check used in Bolander and Blackburn's calculus to handle the language $\mathcal{H}(@, \mathsf{E})$.

We have seen that ignoring the nominals introduced by the rule $(\neg\mathsf{D})$ is crucial to ensure termination. As these nominals only appear as positive literals, their presence does not interfere with the identification of a world with another one. For instance, consider the situation where we have two prefixes $\sigma < \tau$, with their sets of true formulas being $\{p, \Diamond q\}$ and $\{\Diamond q, n\}$ respectively, and n being a new nominal introduced by $(\neg\mathsf{D})$. Here, τ is blocked by σ. It is safe to block like this because it is guaranteed that $\neg n$ never occurs on the branch.

Handling equivalence classes of prefixes: Bolander and Blackburn used the following (Id) rule to handle equivalence classes for $\mathcal{H}(@, \mathsf{E})$:

$$\frac{\sigma a, \tau a, \tau\varphi}{\sigma\varphi}\,(Id)$$

The (Id) rule is an unrestricted version of the (νId) rule. It copies all formulas of an equivalence class to all prefixes of the same equivalence class. This way of handling classes is correct but costly, as it turns out that (νId) alone suffices. The approach of (νId), where information is only copied to the representative prefix of

an equivalence class, is equivalent to the classic disjoint-set forest approach to solve the union-find problem [3].

Kaminski and Smolka do not mention explicitly how to handle equivalence classes. Instead, they make their rules depend on the equational congruence of a branch, that is, the closure of a branch obtained by rewriting every formula and every accessibility statement by replacing every nominal by any other nominal of its equivalence class. For instance in [5], negation is handled by two rules as follows (side conditions are written on the right of each rule):

$$\frac{x \neq y}{\bot}\ x \sim_A y \qquad \frac{(\dot{\neg}p)x}{\bot}\ px \in \tilde{A}$$

with $\tilde{A}$ being the equational congruence of a branch A and $\sim_A$ the least equivalence relation on the nominals of a branch.

In our case, we make explicit the handling of equivalence classes by coping the adequate formulas to representative prefixes, and letting the other rules deal directly with the prefixed formulas present in the branch. In this way our tableau algorithm directly handles equivalence classes. Instanciating Kaminski and Smolka's approach with a disjoint-set forest should yield a very similar system.

Saturation of $(\Diamond)$ and reusing existing accessibility statements: Equivalence classes of prefixes and the loop-check $(\mathcal{I})$ enable us to define a stricter saturation condition for the rule $(\Diamond)$ than in the calculus of Bolander and Blackburn. Indeed, in their calculus, $(\Diamond)$ could be applied on $\sigma\Diamond\varphi$ and then on $\tau\Diamond\varphi$, even when $\sigma \sim \tau$. Here, we take into account the history of applications of $(\Diamond)$ in the whole equivalence class of a given prefix to prevent such redundant diamond expansions. As a consequence, this requires that accessibility statements get copied to the representative of an equivalence class, which is done by (νId).

This has one unfortunate side-effect: our calculus must rely on the loop-check $(\mathcal{I})$ to terminate even for the language $\mathcal{H}(@)$, while Bolander and Blackburn's system doesn't. This is because copying accessibility statements invalidates the argument that prefixes make true smaller and smaller formulas as they are further away from the root prefix. Consider, for instance, the formula

$$n \wedge (\Diamond\top) \wedge (\Box\Diamond\bot) \wedge (\Box n)$$

(with $\Box$ for $\neg\Diamond\neg$), where $(\Diamond)$ is applied systematically before (νId). It does not terminate without the loop-check, but terminates in the system of Bolander and Blackburn.

Pattern-based blocking, loop-check and saturation: The calculus of Kaminski and Smolka relies on pattern-based blocking, which is a restriction on diamond expansions that subsumes both the loop-check $(\mathcal{I})$ and the class-wise saturation condition of $(\Diamond)$. The idea of pattern-based blocking is to only expand a diamond formula if there is no previous diamond expansion in the branch where the created world makes true at least the same formulas. Then, model building is done from a saturated open branch by adding all possible accessibility statements, which includes all those of the blocked diamond formulas.

Pattern-based blocking is a generalisation of the loop-check $(\mathcal{I})$ because it is a

form of anywhere blocking. It is also a generalisation of class-wise saturation, given that it prevents diamond expansions from happening. For these reasons, it would be interesting to integrate this kind of loop-check in our calculus. On the other hand, both pattern-based blocking and $(\mathcal{I})$ fail to ensure termination and completeness for the hybrid logic $\mathcal{H}(@, \Diamond^-)$. Thus, class-wise saturation of $(\Diamond)$ and $(\Diamond^-)$ remains useful in the calculus of $\mathcal{H}(\mathsf{D}, \Diamond^-)$.

4 Adding symmetric and converse modalities

We now consider the language introduced in Section 1, that is, the hybrid language $\mathcal{H}(\mathsf{D}, \Diamond^-)$ with the full signature Sig.

We extend the calculus of $\mathcal{H}(\mathsf{D})$ seen in the previous section so that it handles converse modalities $\Diamond_i^-$ and symmetric modalities. The additional rules are $(\Diamond^-)$, $(\neg\Diamond^-)$, $(bridge)$, (sy), (sy^-) and (tr^-). We call the resulting calculus the calculus of $\mathcal{H}(\mathsf{D}, \Diamond^-)$.

The following example shows that the loop-check $(\mathcal{I})$ pose a problem with converse modalities:

Example 4.1 Consider the unsatisfiable formula $p \wedge \neg\mathsf{E}\neg(\Diamond p \wedge \neg\Diamond^-\Diamond^- p)$. Under restriction $(\mathcal{I})$, a saturated tableau with this formula as root does not close because the first prefix generated by the rule $(\Diamond)$ is blocked by the root prefix. Without $(\mathcal{I})$, we could actually close the tableau by continuing the branch.

In other words, restriction $(\mathcal{I})$ compromises completeness in the presence of converse modalities. We have to define a new restriction that ensures completeness without sacrificing termination.

4.1 Termination

We will establish termination by an argument on infinite chains of prefixes, as in [2] and [5]. To be able to refer to chain of prefixes, we introduce the following relation:

Definition 4.2 If a prefix τ has been introduced in a branch Θ by applying one of the rules $(\Diamond)$ and $(\Diamond^-)$ to a premise $\sigma\varphi$ then we write $\sigma \rhd_\Theta \tau$. We use $\rhd_\Theta^*$ to denote the transitive and reflexive closure of the relation $\rhd_\Theta$.

Saturation by rules $(\Diamond)$ and $(\Diamond^-)$ implies the the following result:

Lemma 4.3 *The graph $(P^\Theta, \rhd_\Theta)$, where P^Θ is the set of prefixes linked by the relation $\rhd_\Theta$, is a forest of finitely branching trees.*

We now prepare the definition of the new loop-check:

Definition 4.4 If σ and τ are two prefixes in a branch Θ such that $L^\Theta(\sigma) = L^\Theta(\tau)$ and not $\sigma \sim_\Theta \tau$, we call them *twins* on Θ.

Definition 4.5 A prefix σ in Θ is said to be *unblocked* if there is no pair of distinct twins τ and τ' such that $\tau \rhd_\Theta^* \tau' \rhd_\Theta^* \sigma$.

Note that if σ is unblocked on Θ and $\sigma' \rhd_\Theta^* \sigma$ then σ' is necessarily also unblocked. The loop-check is defined as follows:

Definition 4.6 (**Loop-check** ($\mathcal{C}$)) The rules ($\Diamond$) and ($\Diamond^-$) are only applied to a formula $\sigma\varphi$ on a branch if σ is unblocked on that branch.

We named this loop-check ($\mathcal{C}$) as in "chain" since this restriction relies on information present in the ancestry chain of a given prefix. We now can prove termination of the calculus of $\mathcal{H}(\mathsf{D}, \Diamond^-)$ with restriction ($\mathcal{C}$):

Theorem 4.7 *Any tableau in the calculus of $\mathcal{H}(D, \Diamond^-)$ constructed under restriction ($\mathcal{C}$) is finite.*

Proof. Suppose there is an infinite tableau. By following the same argument as the one of the proof of Theorem 3.6, we know that there are infinitely many prefixes in the branch, and at the same time, there can be only finitely many applications of the rules (D) and (E). This implies that there are infinitely many applications of ($\Diamond$) and ($\Diamond^-$).

Given Lemma 4.3 and König's lemma, there is one infinite chain of prefixes generated by ($\Diamond$) or ($\Diamond^-$):

$$\sigma_n \rhd_\Theta \sigma_{n+1} \rhd_\Theta \sigma_{n+2} \rhd_\Theta \cdots .$$

Now, there is a maximal number of applications of ($\Diamond$) and ($\Diamond^-$) in a given equivalence class, by definition of the saturation of these rules and the quasi-subformula property. Let us call this number d. Moreover, we know from Lemma 3.2 that there can only be finitely many different sets $L^\Theta(\sigma)$ for σ on the branch Θ. Let m be this number.

Let us consider the prefix $\sigma_{n+d(m+1)+1}$ of the previous chain. It has been introduced by ($\Diamond$) or ($\Diamond^-$) applied on prefix of rank $n + d(m + 1)$ on Θ. Because of restriction ($\mathcal{C}$), $\sigma_{n+d(m+1)}$ must then be unblocked on Θ'. However, there exist two prefixes σ_l and σ_k with $l, k < n + d(m + 1)$, such that $L^\Theta(\sigma_l) = L^\Theta(\sigma_k)$ without $\sigma_l \sim_\Theta \sigma_k$, that is to say σ_l and σ_k are twins. This contradicts $\sigma_{n+d(m+1)}$ being unblocked on Θ', which makes the existence of such an infinite chain impossible. □

4.2 Completeness

In the previous section, inclusion urfathers were used to block other prefixes, and also as elements of the model built from a saturated open branch. But we have seen that a loop-check based on inclusion urfathers is not adequate for completeness in the case of the calculus of $\mathcal{H}(\mathsf{D}, \Diamond^-)$, so we now rely on a weaker loop-check based on unblocked prefixes. Nonetheless, unblocked prefixes cannot be used as elements of an extracted model, since two unblocked prefixes can make true the same nominal. Therefore, we introduce another kind of urfather, the unblocked urfather:

Definition 4.8 Let σ be a prefix occurring in a branch Θ. The *unblocked urfather* of σ on Θ, written $u_\Theta(\sigma)$, is the smallest prefix τ satisfying:

(i) $L^\Theta(\sigma) = L^\Theta(\tau)$

(ii) τ is unblocked.

Such a prefix does not necessarily exist, thus u_Θ is only a partially defined mapping. A prefix σ is called an *unblocked urfather* in Θ if $\sigma = u_\Theta(\tau)$ for some prefix τ.

In other words, the unblocked urfather of a prefix is its smallest unblocked twin. Note that there is no guarantee that it exists, and if it exists, no guarantee that it is on the same chain of ancestry. Thus, a prefix can be blocked without being represented in the possible model.

Note also that the root prefix of a branch Θ will always be an unblocked urfather on that branch. We express that $u_\Theta(\sigma)$ is defined by writing $\sigma \in \mathrm{dom}(u_\Theta)$. We have the following result:

Lemma 4.9 *Let σ be unblocked on a branch Θ. If $\sigma \rhd_\Theta \tau$ then $\tau \in \mathrm{dom}(u_\Theta)$.*

Proof. Assume $\sigma \rhd_\Theta \tau$ where σ is unblocked on Θ. If τ is unblocked on Θ then $\tau \in \mathrm{dom}(u_\Theta)$. So assume that τ is not unblocked. Then there must exist a pair of distinct twins γ, γ' with $\gamma \rhd_\Theta^* \gamma' \rhd_\Theta^* \tau$. Since σ is unblocked we can not have both $\gamma \rhd_\Theta^* \sigma$ and $\gamma' \rhd_\Theta^* \sigma$. Since $\sigma \rhd_\Theta \tau$ this implies $\gamma' = \tau$. Thus τ has γ as a twin, and since necessarily $\gamma \rhd_\Theta^* \sigma$ we get that γ is unblocked. Since γ is a candidate to being the unblocked urfather of τ, $u_\Theta(\tau)$ is defined. □

As in the previous calculus, nominals introduced by $(\neg\mathsf{D})$ are not taken into account when it comes to defining unblocked urfathers. We now prove, as before, the properties Urfather Closure, Urfather Equality and Urfather Characterisation.

Lemma 4.10 (Unblocked Urfather Closure) *Let σ be a prefix in a saturated branch Θ where (νId) is applied, with $\sigma \in \mathrm{dom}(u_\Theta)$ and φ a model-relevant local formula. If $\sigma\varphi \in \Theta$, then $u_\Theta(\sigma)\varphi$.*

Proof. Proof similar to the one of Lemma 3.13. □

Lemma 4.11 (Unblocked Urfather Equality) *Let Θ be a saturated branch. If $\sigma, \tau \in \mathrm{dom}(u_\Theta)$ and $\sigma \sim_\Theta \tau$, then $u_\Theta(\sigma) = u_\Theta(\tau)$.*

Proof. By Lemma 3.7, since $\sigma \sim_\Theta \tau$, then $L^\Theta(\sigma) = L^\Theta(\tau)$, thus $u_\Theta(\sigma) = u_\Theta(\tau)$.□

Lemma 4.12 (Unblocked Urfather Characterisation) *Let Θ be a branch. Then σ is an unblocked urfather if and only if $u_\Theta(\sigma) = \sigma$.*

Proof. For the "only if" direction, suppose σ is an unblocked urfather, i.e., there exists a prefix τ such that $\sigma = u_\Theta(\tau)$. That is, σ is the smallest unblocked prefix such that $L^\Theta(\sigma) = L^\Theta(\tau)$. The prefix $u_\Theta(\sigma)$ has the property of being the smallest unblocked prefix such that $L^\Theta(u_\Theta(\sigma)) = L^\Theta(\sigma)$. As a consequence, $u_\Theta(\sigma)$ is also the unblocked urfather of τ, which means $u_\Theta(\sigma) = \sigma$. □

We are now ready for the model construction that will establish completeness. Given an open, saturated branch Θ with root $\sigma_0\varphi_0$ in the calculus of $\mathcal{H}(\mathsf{D}, \Diamond^-)$, we define a model $\mathcal{M}^\Theta$ by

$$
\begin{aligned}
\mathcal{M}^\Theta &= (W^\Theta, (R_i^\Theta)_{i<n}, V^\Theta), \text{ where} \\
W^\Theta &= \{\sigma \mid \sigma \text{ is an unblocked urfather on } \Theta\} \\
R_i^\Theta &= \{(u_\Theta(\sigma), u_\Theta(\tau)) \mid \sigma\Diamond_i\tau \text{ occurs on } \Theta \text{ and } \sigma, \tau \in \mathrm{dom}(u_\Theta)\} \\
V^\Theta(s) &= \{u_\Theta(\sigma) \mid \sigma s \text{ occurs on } \Theta \text{ and } \sigma \in \mathrm{dom}(u_\Theta)\}.
\end{aligned}
$$

Notice how we define W^Θ: this is because not all prefixes of the branch have an unblocked urfather. That $V^\Theta(a)$ is a singleton for any nominal a follows from Urfather Equality (Lemma 4.11). As before, $\mathcal{M}^\Theta_*$ is the model in which all relations R_i of $\mathcal{M}^\Theta$ are closed respectively for reflexivity, transitivity and symmetry when $i \in \mathcal{R}$, $i \in \mathcal{T}$ and $i \in \mathcal{S}$. We write R_{i*} for the appropriate closure of R_i. We can now prove completeness for the calculus of $\mathcal{H}(\mathsf{D}, \Diamond^-)$ with restriction $(\mathcal{C})$.

Lemma 4.13 *Let Θ be a saturated open branch in the calculus of $\mathcal{H}(D, \Diamond^-)$ with restriction $(\mathcal{C})$ built from the root formula ϕ. For any formula $\sigma\varphi \in \Theta$ such that $\sigma \in \mathrm{dom}(u_\Theta)$ and $nom(\varphi) \subseteq nom(\phi)$, we have $\mathcal{M}^\Theta_*, u_\Theta(\sigma) \models \varphi$.*

Proof. As in the previous completeness proof, the proof is by induction on the syntactic structure of φ:

- φ has one of the forms p, $\neg p$, a, $\neg\neg\psi$, $\psi \wedge \chi$, $\neg(\psi \wedge \chi)$, $\neg\mathsf{E}\psi$, $\neg\mathsf{B}\psi$ or $\neg\Diamond_i\psi$ with i possibly in $\mathcal{R}$ or $\mathcal{T}$. We can directly reuse the previously given proof by simply replacing references to i_Θ by u_Θ and references to "inclusion urfather" by "unblocked urfather". This is because we still have the Urfather Characterisation property (Lemma 4.12).
- $\varphi = \sigma\neg\Diamond_i\psi \in \Theta$ with $i \in \mathcal{S}$. If $(\sigma_1, u_\Theta(\sigma)) \in R^\Theta_{i*}$, then by saturation by (sy), $(bridge)$ and $(\neg\Diamond_i^-)$, $\sigma_1\neg\psi \in \Theta$.
- φ is of the form $\Diamond_i\psi$, $\mathsf{E}\psi$ or $\mathsf{D}\psi$. We can also reuse the previous proof, adding that when a prefix generating rule is applied to the premise $u_\Theta(\sigma)\varphi$ to produce a conclusion $\tau\chi$, then $\tau \in \mathrm{dom}(u_\Theta)$ (Lemma 4.9), which enables us to use the induction hypothesis.
- $\varphi = \Diamond_i^-\psi$. Similar to $\varphi = \Diamond_i\psi$, with adjustments described in the previous case.
- $\varphi = \neg\Diamond_i^-\psi$. If there is no world σ_1 such that $(\sigma_1, u_\Theta(\sigma)) \in R^\Theta_i$ then the property holds trivially. Otherwise, let such σ_1 be chosen arbitrarily. We need to prove that $\mathcal{M}^\Theta, \sigma_1 \models \neg\psi$. By definition of R^Θ_i, there exist prefixes τ, τ_1 such that $\sigma_1 = u_\Theta(\tau_1)$, $u_\Theta(\sigma) = u_\Theta(\tau)$ and $\tau_1\Diamond_i\tau \in \Theta$. By saturation of the $(bridge)$ rule, we have $u_\Theta(\tau_1)\Diamond_i u_\Theta(\tau)$, or in other terms $\sigma_1\Diamond_i u_\Theta(\sigma)$. By Urfather Closure, Θ contains $u_\Theta(\sigma)\neg\Diamond_i^-\psi$, and by saturation of $(\neg\Diamond^-)$, it also contains $\tau\neg\psi$. Induction hypothesis then gives $\mathcal{M}^\Theta, u_\Theta(\sigma_1) \models \neg\psi$, which, by Urfather Characterisation, is equivalent to $\mathcal{M}^\Theta, \sigma_1 \models \neg\psi$.
- $\varphi = \neg\Diamond_i^-\psi$ when $i \in \mathcal{R} \cup \mathcal{T} \cup \mathcal{S}$ is handled as previously, involving the rules $(bridge)$ and (tr^-) and (sy^-) when needed.

□

With a similar argument to the one of Theorem 3.15, we can claim:

Theorem 4.14 *The calculus of $\mathcal{H}(D, \Diamond^-)$ with restriction $(\mathcal{C})$ is complete.*

4.3 Differences with the calculus of $\mathcal{H}(D)$

Normalising accessibility statements: For this calculus, the rule $(bridge)$, which is not present in the calculus of Bolander and Blackburn, is crucial. Indeed, it is now necessary to *normalise* accessibility statements so that they only involve nominal urfathers. Let us use the phrase *forward constraints* to refer to

formulas of the shape $\neg\Diamond\varphi$, *backwards constraints* for formulas of the shape $\neg\Diamond^-\varphi$, and *box constraints* for both shapes.

In the calculus of $\mathcal{H}(\mathsf{D})$, (*bridge*) is not needed because no backwards constraint occur. Forward constraints are propagated along accessibility statements and then copied to nominal urfathers thanks to (νId), as needed. In the calculus of $\mathcal{H}(\mathsf{D}, \Diamond^-)$ without the (*bridge*) rule, a backwards constraint may remain unpropagated. Indeed, by looking at the premises of $(\neg\Diamond^-)$, one can see that nothing will happen if an accessibility statement arrives in the prefix. For instance, the following formula does not yield a closed tableau without the (*bridge*) rule:

$$p \wedge \Diamond n \wedge n\neg\Diamond^- p$$

The calculus of Bolander and Blackburn does not require (*bridge*) because of the way formulas are copied to every element of equivalence classes. In that setting, backwards constraints would first be copied to the adequate prefix before being propagated. Thus (*bridge*) enables us to keep a small footprint of copied formulas.

Loop-check: As in the calculus of Bolander and Blackburn, and the one of Kaminski and Smolka, anywhere blocking cannot be used because it interacts badly with converse modalities. This is why tableaux systems for modal logic and description logics (see, for instance, [6]) that handle converse modalities use a blocking condition using information present only in the ancestry of a given node.

Symmetric relations: Symmetric relations cannot be handled in the calculus of $\mathcal{H}(\mathsf{D})$ shown in Section 3, where the loop-check $(\mathcal{I})$ works by subset checking. The reason lies in the relations that we can build from an open branch, in the presence of this loop-check:

$$R_i^\Theta = \{(\sigma, i_\Theta(\tau)) \mid \sigma \in W^\Theta \text{ and } \sigma\Diamond_i\tau \text{ occurs on } \Theta\}$$

As $i_\Theta(\tau)$ can make true more formulas than τ, it can have more box constraints, thus requiring more information to be present at σ, which is not guaranteed. Inclusion blocking worked well when only forward constraints are present, but here completeness is clearly broken. We find again that blocking and model building using twins is essential for a calculus with converse or with symmetric relations.

5 Discussion and future work

We have presented a prefixed calculus for hybrid logic based on the one presented by Bolander and Blackburn in [2]. We took care of reducing duplication of formulas, while adding support for the difference modality, and reflexive, transitive and symmetric modalities.

Kaminski and Smolka's calculus in [5] is another take on hybrid tableaus. Their calculus is internalised and handles formulas expressed in simple type theory, where equality is natively expressed. Their calculus for $\mathcal{H}(\mathsf{D})$ involves patten-based blocking, a loop-check that subsumes both our loop-check $(\mathcal{I})$ and our class-wise saturation of $(\Diamond^{(-)})$. However, in their system and ours, when moving to the language equipped with converse modalities, a looser loop-check must be used. It seems that a loop-check based on twin detection on a same chain of prefixes is adequate.

In the future, we will explore the use of positive new nominals in the same manner that the difference modality is currently handled, in order to handle functional and injective modalities.

References

[1] F. Baader and U. Sattler. An overview of tableau algorithms for description logics. *Studia Logica*, 69:2001, 2000.

[2] T. Bolander and P. Blackburn. Termination for hybrid tableaus. *Journal of Logic and Computation*, 17(3):517–554, 2007.

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2001.

[4] M. Kaminski and G. Smolka. Hybrid tableaux for the difference modality. *Electron. Notes Theor. Comput. Sci.*, 231:241–257, 2009.

[5] M. Kaminski and G. Smolka. Terminating tableau systems for hybrid logic with difference and converse. *Journal of Logic, Language and Information*, 18(4):437–464, 2009.

[6] D. Tsarkov, I. Horrocks, and P. F. Patel-Schneider. Optimizing terminological reasoning for expressive description logics. *J. of Automated Reasoning*, 39(3):277–316, 2007.